

Figure 1 (Prior art)

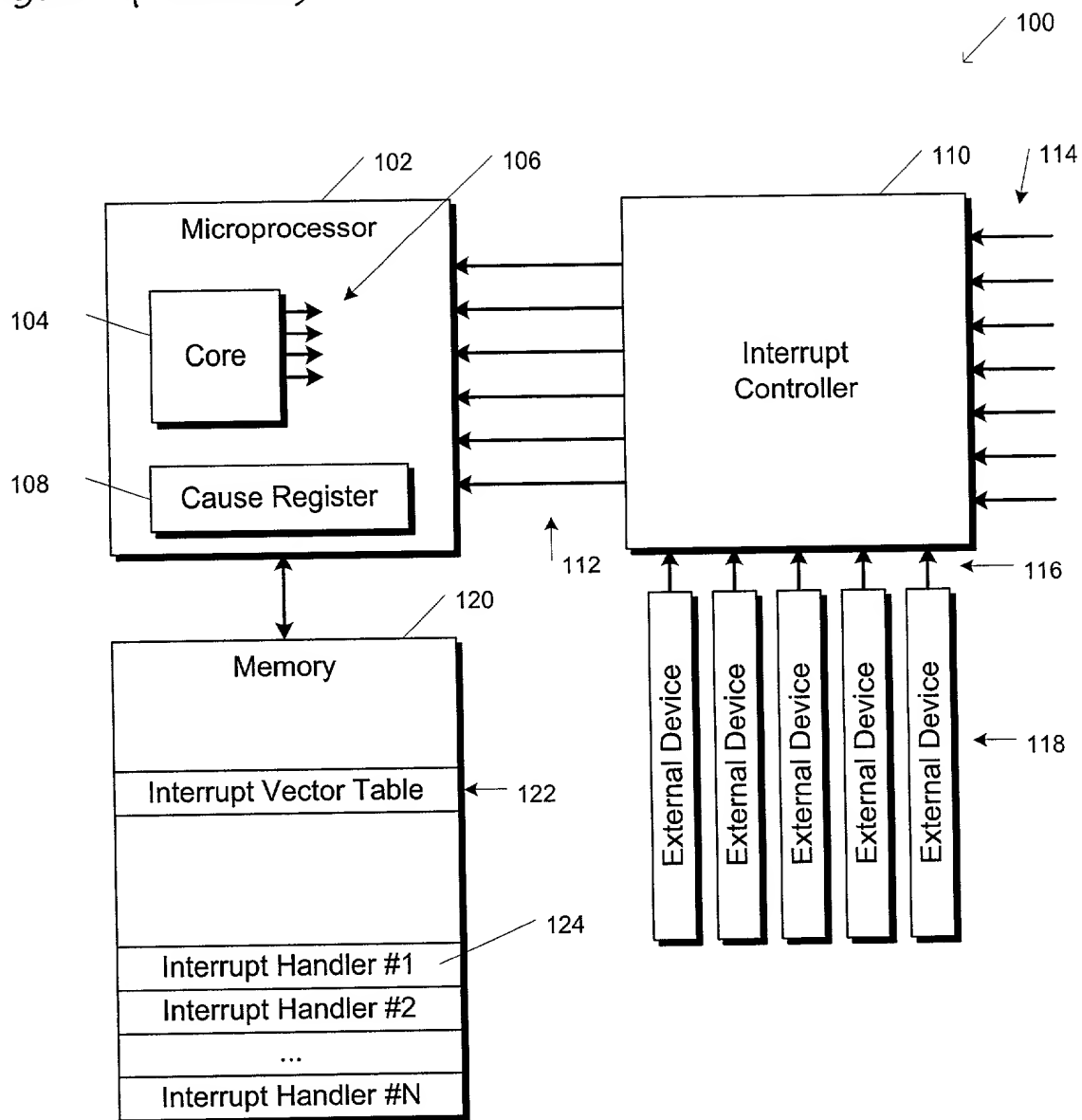


Figure 2 (Prior Art)

Flow Chart for Prior Art Interrupt Handling

200

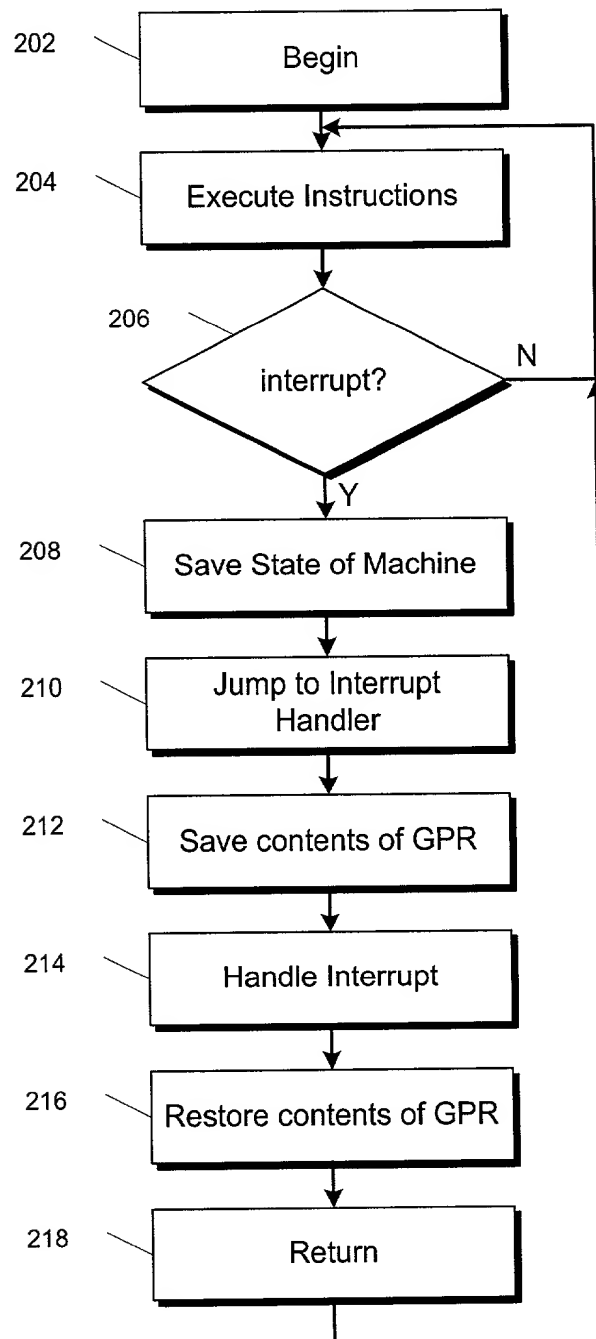


Figure 3

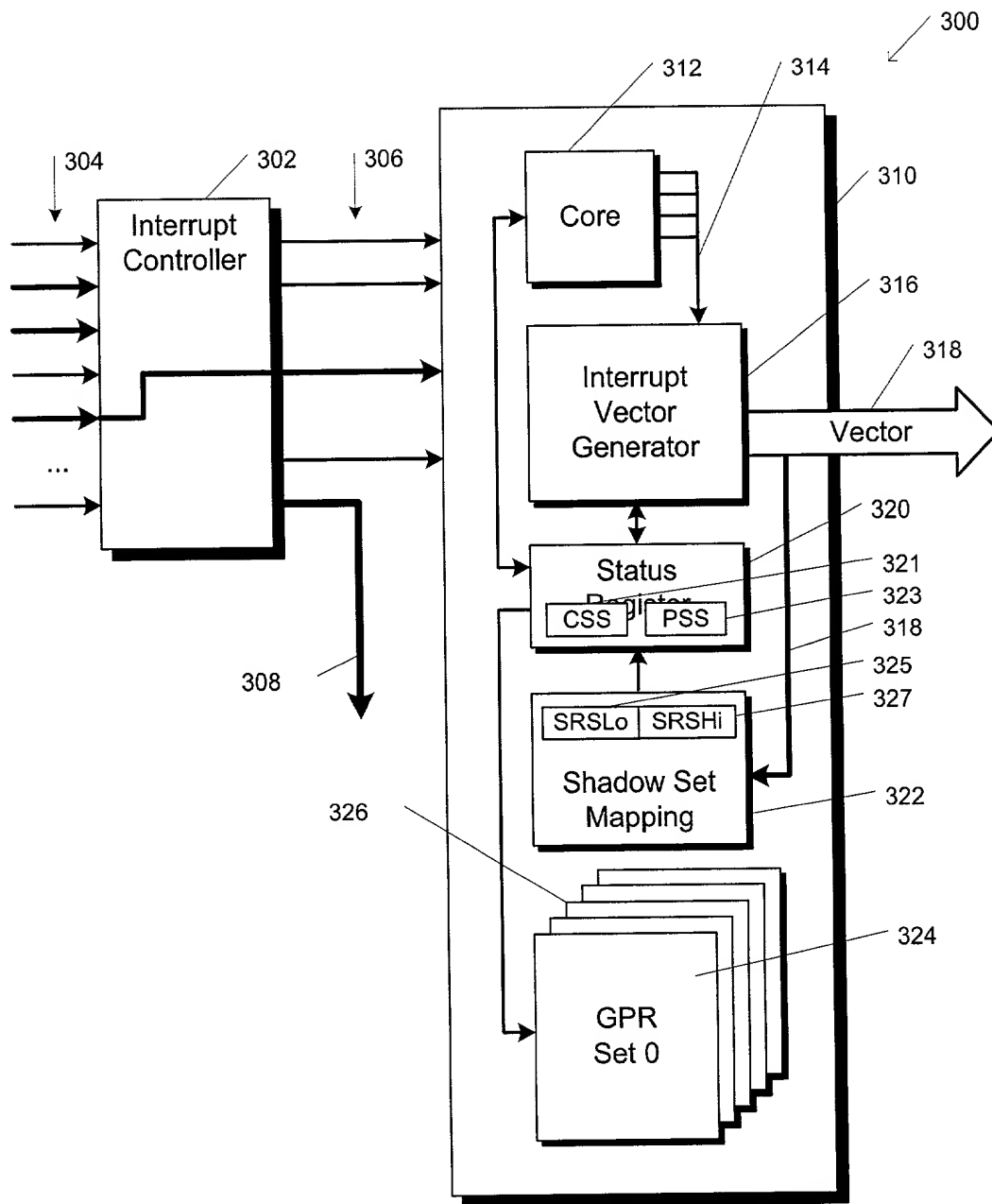


Figure 4

400

General Purpose Registers

| Register Number | Name | Used For |
|-----------------|-------|---|
| 0 | zero | Always returns 0 |
| 1 | at | (assembler temporary) Reserved for use by assembler |
| 2-3 | v0,v1 | Value returned by subroutine |
| 4-7 | a0-a3 | (arguments) First few parameters in a subroutine |
| 8-15 | t0-t7 | (temporaries) Subroutines can use without saving |
| 24,25 | t8,t9 | |
| 16-23 | a0-a7 | Subroutine register variables; a subroutine that writes one of these must save the old value and restore it before it exits, so the calling routine sees the values preserved |
| 26,27 | k0,k1 | Reserved for use by interrupt/trap handler |
| 28 | gp | Global pointer |
| 29 | sp | Stack pointer |
| 30 | s8/fp | Ninth register variable; subroutines that need one can use this as a frame pointer |
| 31 | ra | Return address for subroutine |

Figure 5

500

Status Register Format - Status1

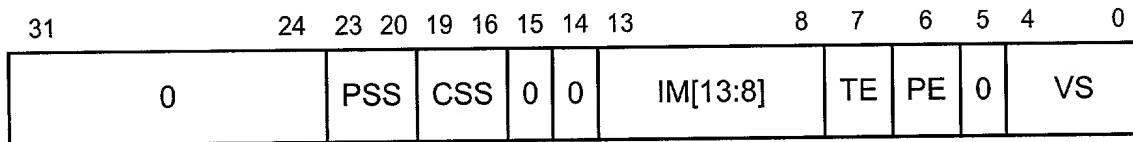


Figure 7

700

SRSLo Register Format

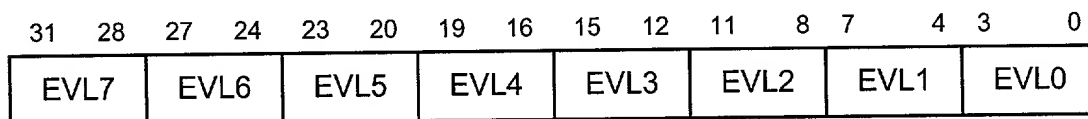


Figure 9

900

SRSHi Register Format

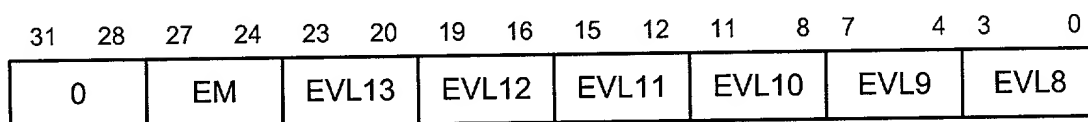


Figure 6a

600

Status1 Register Field Descriptions

| Fields | | Description | Read/ Write | | | | | | |
|----------|------------------------------------|--|----------------|---------|---|---------------------------------|---|------------------------------------|-----|
| Name | Bits | | | | | | | | |
| 0 | 31..204 15..14, 5 | Must be written as zero; returns zero on read | 0 | | | | | | |
| PSS | 23..20 | <p>Previous Shadow Set. If GPR shadow registers are implemented, this field is copied from the CSS field when an exception or interrupt occurs and provides the value of CSS. An eret instruction copies this value back into the CSS field.</p> <p>If no GPR shadow registers are implemented, this field is ignored on writes and returns zero on read.</p> | R/W | | | | | | |
| CSS | 19..16 | <p>Current Shadow Set. If GPR shadow registers are implemented, this field is the number of the current GPR set.</p> <p>If no GPR shadow registers are implemented, this field is ignored on write and returns zero on read.</p> | R/W | | | | | | |
| IM[13:8] | 13..8 | <p>Interrupt Mask: Controls the enabling of the IP[13:8] interrupts.</p> <table><tr><th>Encoding</th><th>Meaning</th></tr><tr><td>0</td><td>Interrupt request disabled</td></tr><tr><td>1</td><td>Interrupt request enabled</td></tr></table> | Encoding | Meaning | 0 | Interrupt request disabled | 1 | Interrupt request enabled | R/W |
| Encoding | Meaning | | | | | | | | |
| 0 | Interrupt request disabled | | | | | | | | |
| 1 | Interrupt request enabled | | | | | | | | |
| TE | 7 | <p>Timer Exclusive. This bit determines whether the timer interrupt is combined in an implementation dependent way with hardware interrupt 5 to become IP5 (default) or redirected to IP12.</p> <table><tr><th>Encoding</th><th>Meaning</th></tr><tr><td>0</td><td>Timer interrupt combined as IP5</td></tr><tr><td>1</td><td>Timer interrupt redirected to IP12</td></tr></table> | Encoding | Meaning | 0 | Timer interrupt combined as IP5 | 1 | Timer interrupt redirected to IP12 | R/W |
| Encoding | Meaning | | | | | | | | |
| 0 | Timer interrupt combined as IP5 | | | | | | | | |
| 1 | Timer interrupt redirected to IP12 | | | | | | | | |

Figure 6b

600

Status1 Register Field Descriptions, cont.

| Fields | | Description | Read/ Write | | | | | | | | | | | | | | | | |
|--------------------------------|------------------------------------|---|----------------|-----------------------------------|-------|---------------------------------|-------|------------------------------------|-------|----|-------|-----|-------|-----|-------|-----|--------------------------------|--|-----|
| Name | Bits | | | | | | | | | | | | | | | | | | |
| PE | 6 | <p>Performance Counter Exclusive. This bit determines whether the performance counter interrupt is combined in an implementation dependent way with hardware interrupt 5 to become IP5 (default) or redirected to IP13.</p> <table><tr><th>Encoding</th><th>Meaning</th></tr><tr><td>0</td><td>Timer interrupt combined as IP5</td></tr><tr><td>1</td><td>Timer interrupt redirected to IP13</td></tr></table> | Encoding | Meaning | 0 | Timer interrupt combined as IP5 | 1 | Timer interrupt redirected to IP13 | R/W | | | | | | | | | | |
| Encoding | Meaning | | | | | | | | | | | | | | | | | | |
| 0 | Timer interrupt combined as IP5 | | | | | | | | | | | | | | | | | | |
| 1 | Timer interrupt redirected to IP13 | | | | | | | | | | | | | | | | | | |
| VS | 4..0 | <p>Vector Spacing. This field specifies the spacing between vectored interrupts and exceptions.</p> <table><tr><th>Encoding</th><th>Spacing Between Vectors (decimal)</th></tr><tr><td>16#00</td><td>0</td></tr><tr><td>16#01</td><td>32</td></tr><tr><td>16#02</td><td>64</td></tr><tr><td>16#04</td><td>128</td></tr><tr><td>16#08</td><td>256</td></tr><tr><td>16#10</td><td>512</td></tr><tr><td colspan="2">All other values are reserved.</td></tr></table> | Encoding | Spacing Between Vectors (decimal) | 16#00 | 0 | 16#01 | 32 | 16#02 | 64 | 16#04 | 128 | 16#08 | 256 | 16#10 | 512 | All other values are reserved. | | R/W |
| Encoding | Spacing Between Vectors (decimal) | | | | | | | | | | | | | | | | | | |
| 16#00 | 0 | | | | | | | | | | | | | | | | | | |
| 16#01 | 32 | | | | | | | | | | | | | | | | | | |
| 16#02 | 64 | | | | | | | | | | | | | | | | | | |
| 16#04 | 128 | | | | | | | | | | | | | | | | | | |
| 16#08 | 256 | | | | | | | | | | | | | | | | | | |
| 16#10 | 512 | | | | | | | | | | | | | | | | | | |
| All other values are reserved. | | | | | | | | | | | | | | | | | | | |

FOOT-48022560

Figure 8

800

SRSLo Register Field Descriptions

| Fields | | Description | Read/ Write |
|--------|--------|---|----------------|
| Name | Bits | | |
| EVL7 | 31..28 | Shadow register set number for Exception Vector Level 7 | R/W |
| EVL6 | 27..24 | Shadow register set number for Exception Vector Level 6 | R/W |
| EVL5 | 23..20 | Shadow register set number for Exception Vector Level 5 | R/W |
| EVL4 | 19..16 | Shadow register set number for Exception Vector Level 4 | R/W |
| EVL3 | 15..12 | Shadow register set number for Exception Vector Level 3 | R/W |
| EVL2 | 11..8 | Shadow register set number for Exception Vector Level 2 | R/W |
| EVL1 | 7..4 | Shadow register set number for Exception Vector Level 1 | R/W |
| EVL0 | 3..0 | Shadow register set number for Exception Vector Level 0 | R/W |

Figure 10

1000

SRSHi Register Field Descriptions

| Fields | | Description | Read/ Write |
|--------|--------|--|----------------|
| Name | Bits | | |
| 0 | 31..28 | Must be written as zero; returns zero when read | R/W |
| EM | 27..24 | Shadow register set number for non-vectorred Exception Mode (Status _{EXL} =1). This value is used only if an interrupt is not serviced thru the vectored exception table. | R/W |
| EVL13 | 23..20 | Shadow register set number for Exception Vector Level 13 | R/W |
| EVL12 | 19..16 | Shadow register set number for Exception Vector Level 12 | R/W |
| EVL11 | 15..12 | Shadow register set number for Exception Vector Level 11 | R/W |
| EVL10 | 11..8 | Shadow register set number for Exception Vector Level 10 | R/W |
| EVL9 | 7..4 | Shadow register set number for Exception Vector Level 9 | R/W |
| EVL8 | 3..0 | Shadow register set number for Exception Vector Level 8 | R/W |

Figure 1 consists of 12 subplots, labeled (a) through (l), each showing a different physiological or behavioral parameter over a 12-hour period. The x-axis for all plots represents time in hours, from 0 to 12. The y-axis represents the value of the parameter. The parameters are: (a) HR (Heart Rate), (b) HRV (Heart Rate Variability), (c) HRV, (d) HRV, (e) HRV, (f) HRV, (g) HRV, (h) HRV, (i) HRV, (j) HRV, (k) HRV, and (l) HRV. Each plot shows a line graph with error bars, indicating the mean and standard deviation of the data. The data points are connected by lines, showing the trend over time. The parameters generally show a decrease or fluctuation over the 12-hour period.

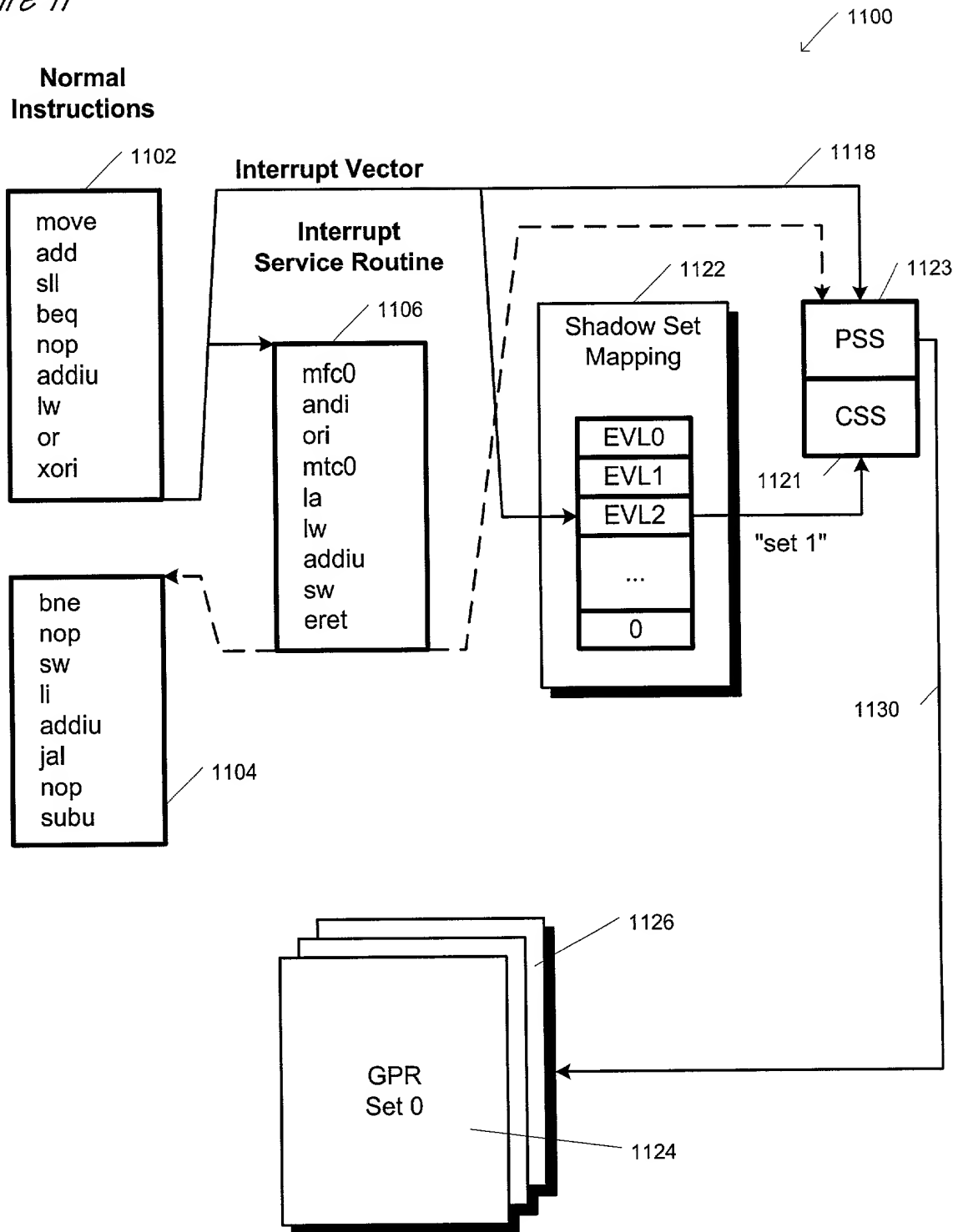


Figure 12

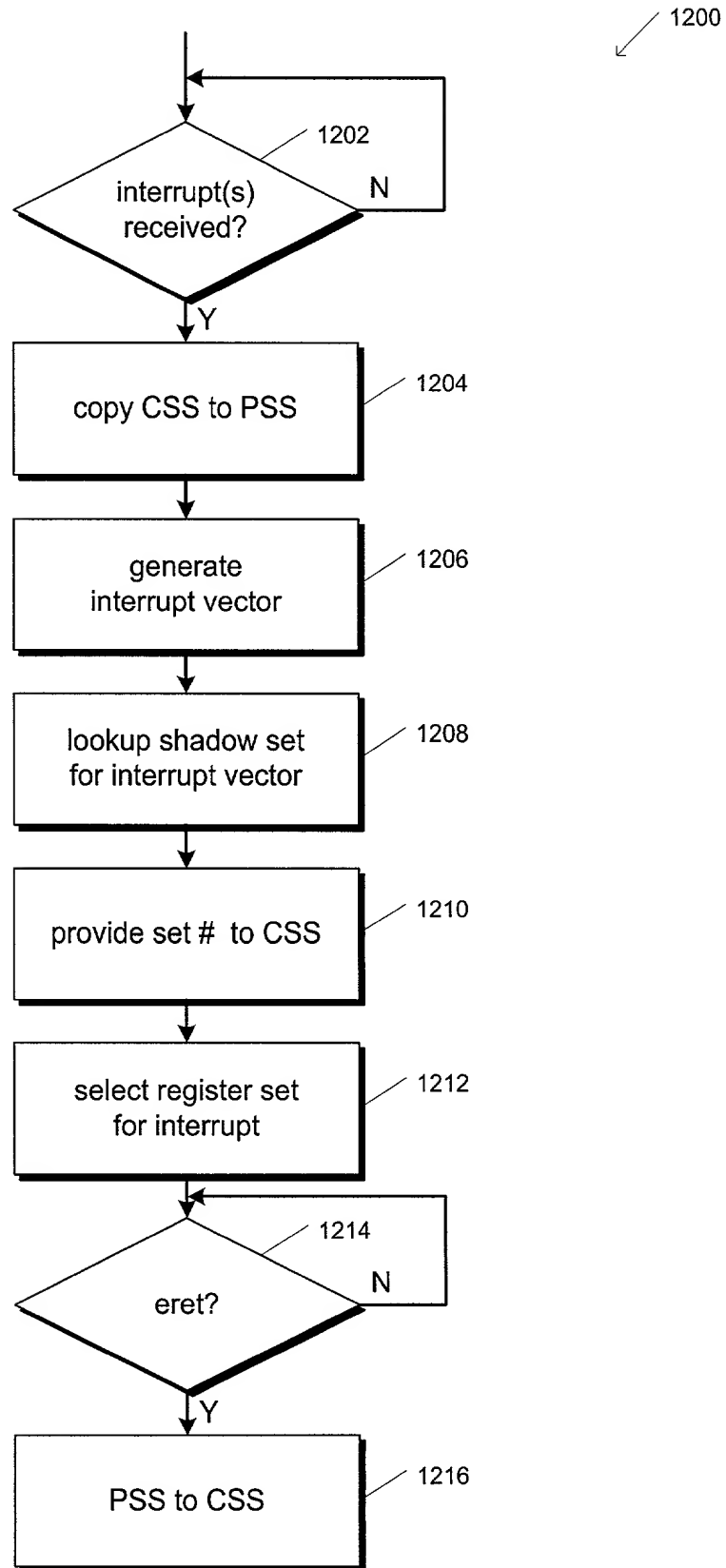
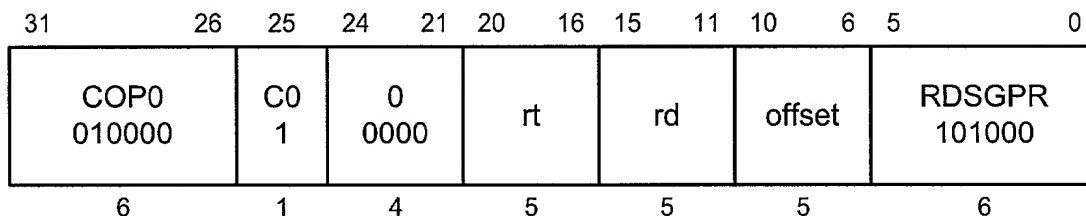


Figure 13

1300

Read Shadow GPR Instruction - RDSGPR



Format: RDSGPR rt, offset(rd)

Purpose:

To move the contents of a shadow GPR register to a current GPR.

Description: $rt \leftarrow SGPR[rd + offset]$

The contents of the shadow GPR register specified by the sum of offset and rd is moved to the current GPR rt.

Restrictions:

The results are UNDEFINED if the sum of offset and rd do not reference an implemented shadow register.

Operation:

```

if IsCoprocessorEnabled(0) then
    if (ArchitectureRevision >= 2) then
        GPR[rt] ← SGPR[rd + offset]
    else
        SignalException(ReservedInstruction)
    endif
else
    SignalException(CoprocessorUnusable, 0)
endif
    
```

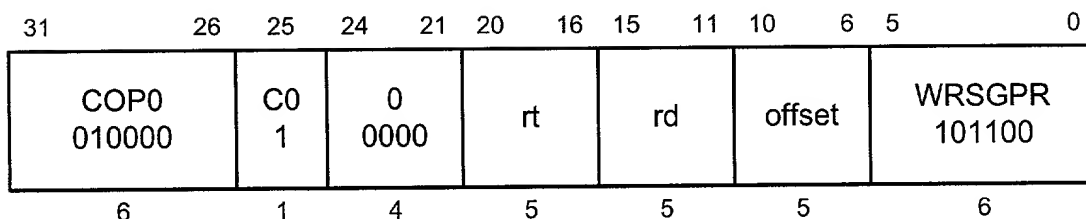
Exceptions:

Coprocessor Unusable
Reserved Instruction

Figure 14

1400

Write Shadow GPR Instruction - WRSGPR



Format: WRSGPR rt, offset(rd)

Purpose:

To move the contents of a current GPR to a shadow GPR.

Description: SGPR [rd + offset] ← rt

The contents of the current GPR rt is moved to the shadow GPR register specified by the sum of offset and rd.

Restrictions:

The results are UNDEFINED if the sum of offset and rd do not reference an implemented shadow register.

Operation:

```

if IsCoproprocessorEnabled(0) then
    if (ArchitectureRevision >= 2) then
        SGPR[rd = offset] ← GPR[rt]
    else
        SignalException(ReservedInstruction)
    endif
else
    SignalException(CoproprocessorUnusable, 0)
endif
    
```

Exceptions:

Coproprocessor Unusable
Reserved Instruction